بسم الله الرحمن الرحيم

و الصلاة و السلام على حبيبنا محمد صلى الله عليه

و على آله أجمعين الى يوم الدين

السلام عليكم و رحمة الله و بركاته

التعامل مع الملفات بالدلفي

يمكن أن نصنف الملفات إلى 3 أقسام :

1- الملفات النصية : تتكون من سطر أو عدة أسطر أفقية لها خصائص محددة كنهاية السطر و موضع المؤشر فيه تحتوي على معلومات ذات نمط سلاسل محرفية كما يمكن الرجوع أو القفز من و إلى السطر الموالي في النص حسب موضع المؤشر , كما تعتبر أسهل انوع الملفات من حيث التعامل معها (كتابة , قراءة , حذف .... إلخ) .

2- الملفات التتابعية : تعتمد على مبدأ مختلف من حيث بنيتها عن الصنف الأول .. هنا تتكون من سجل أو مجموعة سجلات ذات عدة حقول كجداول البيانات لكنها لا تملك القدرة على تخزين كل أنماط المعلومات وهي 3 أنماط : الصنف , المؤشرات , و الجداول الديناميكية إذا فهي محدودة لكن نقطة قوتها في كتابة أو قراءة حقول كاملة لن نستطيع تحقيقها بالملفات النصية .

3- الملفات الثنائية : ملفات من دون نمط تتيح لنا كتابة أو قراءة كل أنواع المعطيات مهما كان نمطها بطريقة قوية و آمنه و بكل حرية مئة بالمـــئة بغض النظر على إمتداد الملف .

1- الملفات النصية : كيف ننشأ ملف نصي بسيط ؟؟ كبداية نستعمل الإجراء :

```
procedure AssignFile ( var FileHandle : TextFile; const FileName : string ) ;
```

كما نلاحظ الإجراء AssignFile يمكننا من إرفاق مقبض الملف نصي أو ثنائي الى مسار ملف جدية نود إنشائه أو موجود سابقا نود الكتابة فيه أو القرائة منه لكن قبل ذلك يتوجب علينا تحديد طريقة فتح الملف او كيف نريده أن يفتح :

- لنفتح ملف للقراءة فقط نستعمل الإجراء ReSet يجب أن يكون الملف موجود مسبقا و إلى سيقع خطأ أثناء التنفيذ (أما موضوع معالجة الإخطاء و تسييرها و الإعتماد على توجيهات المترجم فهي درس في حد ذاته) .

```
procedure Reset ( var FileHandle : TextFile ) ;
```

- لنفتح ملف للكتابة فقط نستعمل الإجراء ReWrite لا يهم شرط وجود الملف لكن يجب أن يكون المسار صحيح أو منطقي في جميع الحالات .. في حالة وجود الملف سيتم حذف محتواه و الكتابة عليه أما إذا كان غير موجود سيتم إنشاء ملف جديد .

```
procedure ReWrite ( var FileHandle : TextFile ) ;
```

- لكتابة قيمة نصية String نستعمل الإجراء Write للكتابة في السطر الحالي أو WriteLn للكتابة في نفس السطر مع إرجاع المؤشر للسطر الموالي .

```
1- procedure Write ( var FileHandle : TextFile; Expression1,Expression2) ;

2- procedure Write ( var FileHandle : TextFile; var Value) ;
```

- لقراءة قيمة نصية String نستعمل الإجراء Read لقراءة المعلومة وفق موضع المؤشر في السطر الحالي او ReadLn لقراءة المعلومة الأولى في السطر مع إرجاع المؤشر للسطر الموالي .

```
procedure Read ( var FileHandle : TextFile; var Value) ;
```

- لتحرير الملف بعد إنشائه :

```
procedure CloseFile ( var FileHandle : TextFile) ;
```

*** أمثلة بسيطة :**

```
--كيف نكتب في ملف نصي----------------------------------------------------------------

Var F:TextFile; //TextFile متغير من نوع
Begin
 AssignFile(F,'C:\Test.txt') ;// نرفق المتغير بالمسار الصحيح إفتراضيا
 ReWrite(F) ;//نفتح الملف بخاصية للكتابة فقط
 Write(F,'Samle Text ... ') ;//نكتب نص بسيط
 CloseFile(F) ;// نحرر الملف
End;

--كيف نقرأ من ملف نصي---------------------------------------------------------------

Var F:TextFile;
    Temp:String;
Begin
If Not FileExists('C:\Test.txt') Then Exit;// نتحقق أولا من وجود الملف
AssignFile(F,'C:\Test.txt');
ReSet(F);// نعطي الملف في حالة وجوده الخاصية للقرائة فقط
While Not Eof (F) Do// نفتح حلقة تكرارية بشرط أن الملف غير منتهي
 Begin
 Read(F,Temp);// Temp نقراء المعلومة في المتغير
 ShowMessage(Temp);//نعرض القيمة المقروئة في رسالة
 End;
CloseFile(F);
End;

Eof : End Of File نهاية الملف
FileExists : دالة للتحقق من شرط وجود ملف في المسار المدخل
```

2- **الملفات التتابعية** : هنا فقط سنتعامل مع سجلات ذات حقول بدل الأسطر لذا سنحتاج لدوال و إجراءات بسيطة تسهل علينا التعامل مع هذا النوع من الملفات .

- لننشأ نمط جديد ذو حقول مختلفة لتعميم التعامل معها بكل بساطة Record/Packed.
- يمكن أن نستعمل توجيهات المترجم هنا للتقليل من حجم النمط {$A+}/{$A-}.
- كما عرفنا من قبل فإن الجداول الديناميكية غير مسموحة و عليه يجب تحديدها .
- الإجراء Seek للإزاحة أو تموضع المؤشر في الملف كما سنستعمل الإجراء TrunCate لبتر أو مسح محتوى الملف إنطلاقا من موضع المؤشر الحالي .

الأن كيف نكتب أو نقرأ قيمة مستغلين بذالك خصائص هذا النوع من الملفات :

* أمثلة بسيطة :

```
--كيف ننشأ سجل Registration------------------------------------------------------

{$A-}
Type
 MyType = Record// نعرف سجل جديد ذو عدة حقول
  Name:String[100];//حددنا طول السلسلة بـ 100 حرف
  Age:Byte;// هل عمر الإنسان فوق 256 عام ؟؟؟
  Men:Boolean;// إذا كان خطأ يعني إنثى
 End;
FRec = File Of MyType;//هنا عرفنا F على أنه ملف من نوع MyType الذي يملك عدة حقول

{Or   // {$A-} = Packed Record .// SizeOf(MyType) قمنا بضغط حجم الريكورد جرب
Type
 MyType = Packed Record
  Name:String[100];
  Age:Byte;
  Men:Boolean;
 End;
FRec = File Of MyType;}

--كيف نكتب في الملف التتابعي------------------------------------------------------

Var Temp:MyType;
    F:FRec;
Begin
AssignFile(F,'C :\Test.Rec');// لا يهم إمتداد الملف
If FileExists('C:\Test.Rec')Then// إذا كان الملف موجود يفتح للقراءة و العكس للكتابة
ReSet(F) Else ReWrite(F);//لا يهم طريقة فتح الملف بخاصية القرائة أو الكتابة
Temp.Name:='TF6M';// نقوم بملأ الحقول
Temp.Age:=20;
Temp.Men:=True;
Write(F,Temp);// نكتب السجل Temp بعد أن ملائنا حقول السجل
CloseFile(F);
End;
```

```
--كيف نقرأ من ملف تتابعي-------------------------------------------------------

Var Temp:MyType;
    F:FRec;
Begin
AssignFile(F,'C:\Test.Rec');
If FileExists('C:\Test.Rec')Then
ReSet(F) Else ReWrite(F);
While Not Eof (F) Do
 Begin
 Read(F,Temp);// نقرأ السجل
 ShowMessage('My Name: '+Temp.Name+#13+
             'My Age: '+IntToStr(Temp.Age));// نعرض النتيجة في رسالة بسيطة
 If Temp.Men Then
   ShowMessage('I am Men')Else
   ShowMessage('I am Women');
 End;
CloseFile(F);// نحرر الملف
End;

--كيف نكتب في ملف تتابعي مع مسح المعلومات المكتوبة سابقا----------------------------

Var Temp:MyType;
    F:FRec;
Begin
AssignFile(F,'C :\Test.Rec');// لا يهم إمتداد الملف
If FileExists('C:\Test.Rec')Then// إذا كان الملف موجود يفتح للقراءة و العكس للكتابة
ReSet(F) Else ReWrite(F);//لا يهم طريقة فتح الملف بخاصية القرائة أو الكتابة
Seek(F,0) ;//نضع المؤشر في بداية الملف الموضع 0
TrunCate(F) ;//مسح جميع مدونات الملف إنطلاقا من موضع المؤشر
Temp.Name:='TF6M';// نقوم بملأ الحقول
Temp.Age:=20;
Temp.Men:=True;
Write(F,Temp);// نكتب السجل Temp بعد أن ملائنا حقول السجل
CloseFile(F);
End;
```

3- **الملفات الثنائية** : و هي أهم انواع الملفات لأنها تتيح لنا الحرية الكاملة لكتابة أو قراءة جميع انواع البيانات ... فقط تحتاج معاملة خاصة و سأحاول تنويع طرق التعامل معها بكل بساطة .

- هناك من يفضل إستخدام Api بدل الدوال المسبلة في الدلفي :

CreateFile = AssignFile
ReSet/ReWrite = ReadFile/WriteFile
Seek = SetFilePointer

- كما يمكن ان نستغل دالة IoResult بواسطة توجيه المترجم {$I-}/{$I+} بدل الدالة FileExists لتجنب نظافة الوحدة SysUtils للتقليل من حجم البرنامج .
- هنا نستعمل BlockWrite/BlockRead مكان Read/Write .

```
procedure BlockRead ( var FileHandle : File; var Buffer; RecordCount:Integer) ;

procedure BlockWrite ( var FileHandle : File; var Buffer;RecordCount:Integer) ;
```

الأن كيف نكتب أو نقرأ قيمة مستغلين بذالك خصائص هذا النوع من الملفات :

* أمثلة بسيطة :

```
{**************************************}
          //Open BinaryFile//
{**************************************}

Function OpenBFile(Var F:File;Path:String):Boolean;
 Begin
 {$I-}//إبطال معالجة الأخطاء الألية
 AssignFile(F,Path);
 If FileExists (Path) Then ReSet(F,1) Else ReWrite (F,1);
 Result:=(IoResult = 0);
 If Not Result Then Exit;
 Seek(F,0);
 {$I+}// إعادة تفعيل خاصية معالجة الاخطاء
 End;
{Or

Function OpenBFile(Var F:File Of Byte ;Path:String):Boolean;
 Begin
 {$I-}//إبطال معالجة الأخطاء الألية
 AssignFile(F,Path);
 If FileExists (Path) Then ReSet(F) Else ReWrite (F);
 Result:=(IoResult = 0);// إذا كانت القيمة المرجعه 0 يعني لا يوجد أخطاء
 If Not Result Then Exit;// في حالة وجود خطأ نخرج من الأجراء
 Seek(F,0);
 {$I+}// إعادة تفعيل خاصية معالجة الاخطاء
 End;}

{**************************************}
          //Close BinaryFile//
{**************************************}

Function CloseBFile(Var F:File):Boolean;
 Begin
 {$I-}
 CloseFile(F);
 Result:=(IoResult = 0);
 If Not Result Then Exit;
 {$I+}
 End;

{**************************************}
       //Write String Value//
{**************************************}

Function WriteSBFile(BPath:String;LS:String):Boolean;
Var F:File;
    LongWs, NBwrite: Integer;
 Begin
 If Not OpenBFile (F,BPath) Then Exit;
 {$I-}
 TrunCate(F);// مسح الملف
 If IoResult <> 0 Then Exit;// إذا كان هناك خطأ نخرج من الدالة
 LongWs:=Length(LS);// LongWs = LS طول السلسلة النصية التي نريد كتابتها
 BlockWrite(F,LongWs,SizeOf(LongWs),NBwrite);//أول شيء نكتبه هو طول السلسلة
 If (IoResult <> 0) Or (SizeOf(LongWs)<>NBwrite) Then Exit;
 BlockWrite(F,LS[1],LongWs,NBwrite);// يكفي كتابة الحرف الاول من السلسلة و الباقي
يسجل آليا .
 If (IoResult <> 0) Or (LongWs <> NBwrite) Then Exit;// في حالة وجود خطأ أو القيمة
المكتوبة تختلف على الحجم الفعلي للسلسة نخرج من الدالة .
 {$I+}
```

```
 Result:=CloseBFile(F);
 End;

{**************************************}
       //Write ShortString Value//
{**************************************}

Function WriteCsBFile(BPath:String;CS:ShortString):Boolean;
Var F:File;
    NBwrite: Integer;
 Begin
 If Not OpenBFile (F,BPath) Then Exit;
 {$I-}
 TrunCate(F);
 If IoResult <> 0 Then Exit;
 BlockWrite(F,CS[0],Byte(CS[0])+1,NBwrite);
 If (IoResult <> 0) Or (Byte(CS[0])+1 <> NBwrite) Then Exit;
 {$I+}
 Result:=CloseBFile(F);
 End;
// هنا لا نحتاج تسجيل طول السلسلة لأنها محصورة بين 0 و 256 محرف
//يكفي قراءة الـ CS[0] التي تحتوي على طول السلسلة
//كما نضيف Byte ليسجل فيه طول السلسلة المقروء

{**************************************}
         //Read String Value//
{**************************************}

Function ReadSBFile(BPath:String;LS:String):Boolean;
Var F:File;
    LongRs, NBwrite :Integer;
 Begin
 If Not OpenBFile (F,BPath) Then Exit;
 {$I-}
 If IoResult <> 0 Then Exit;
 BlockRead(F,LongRs,4,NBwrite);// 4 = SizeOf (String);
 If (IoResult <> 0) Or (4 <> NBwrite) Then Exit;
 SetLength(Ls,LongRs);
 BlockRead(F,Ls[1],LongRs,NBwrite);
 If (IoResult <> 0) Or (LongRs <> NBwrite) Then Exit;
 {$I+}
 Result:=CloseBFile(F);
 End;

{**************************************}
         //Read ShortString Value//
{**************************************}

Function ReadCsBFile(BPath:String;CS:ShortString):Boolean;
Var F:File;
    LongRc:Byte;
    NBwrite: Integer;
 Begin
 If Not OpenBFile (F,BPath) Then Exit;
 {$I-}
 If IoResult <> 0 Then Exit;
 BlockRead(F,LongRc,1,NBwrite);// 1 = SizeOf (ShortString);
 If (IoResult <> 0) Or (1 <> NBwrite) Then Exit;
 SetLength(CS,LongRc);
 BlockRead(F,CS[1],LongRc,NBwrite);
 If (IoResult <> 0) Or (LongRc <> NBwrite) Then Exit;
 {$I+}
 Result:=CloseBFile(F);
 End;
```

شرح إجراء الكتابة :

```
BlockWrite ( var FileHandle : File; var Buffer; RecordCount : Integer {; var RecordsRead
: Integer} ) ;
```

FileHandle // ملف بدون نمط

Buffer // مساحة من الذاكرة غير محددة النمط يمكن أن نسجل بها أي قيمة لكتابته في الملف

RecordCount // حجم المعلومة التي نريد ان نكتبها

RecordsWritten // بارامتر إختياري يسجل الحجم الحقيقي المكتوب

شرح إجراء القراءة :

```
BlockRead ( var FileHandle : File; var Buffer; RecordCount : Integer {; var RecordsRead
: Integer} ) ;
```

FileHandle // ملف بدون نمط

Buffer //مساحة من الذاكرة غير محددة النمط تأخذ قيمة المعلومة المقروءة

RecordCount // حجم المعلومة التي نريد ان نقرأها

RecordsWritten // بارامتر إختياري يسجل الحجم الحقيقي المقروء

* كيف ننشأ ملف ثنائي فقط بدوال Api أي بوحدة Windows.Pas فقط ؟؟؟ :

ملاحظة : تم الإستعانة بشرح التالي فقط لإختصار الوقت : http://www.arabteam2000-forum.com/lofiversion/index.php/t150145.html

أو يمكنك مراجعة المصدر الرسمي لميكروسوفت : http://msdn.microsoft.com/en-us/library/aa363858(VS.85).aspx

1- الدالة CreateFile :

```
Function CreateFile(lpFileName: PChar; dwDesiredAccess, dwShareMode: DWORD;
  lpSecurityAttributes: PSecurityAttributes; dwCreationDisposition,
dwFlagsAndAttributes: DWORD;
  hTemplateFile: THandle): THandle; stdcall;
```

lpFileName : اسم الملف أو المنفذ المراد فتحه او انشاؤه .

dwDesiredAccess : يحدد العملية المراد اجرائها على الملف :

1- GENERIC_READ : يتم فتح الملف أو المنفذ للقراءة منه فقط .
2- GENERIC_WRITE : يتم فتح الملف أو المنفذ للكتابة فيه (لإنشاء ملف جديد) .
3- GENERIC_ALL : يتم فتح الملف أو المنفذ للكتابة و القراءة و للتنفيذ معا .
4- GENERIC_Execute : يتم فتح الملف للتنفيذ فقط .

dwShareMode : لتحديد العمليات القابلة للتنفيذ على الملف من قبل البرامج الأخرى :

1- FILE_SHARE_READ : يتم قبول عمليات القراءة من الملف أو المنفذ من برامج اخرى.
2- FILE_SHARE_WRITE : يتم قبول عمليات الكتابة الى الملف أو المنفذ من برامج اخرى.
3- FILE_SHARE_DELETE : يتم قبول عملية مسح الملف أو المنفذ من برامج اخرى.

4- FILE_SHARE_ALL : يتم قبول كل عمليات القراءة و الكتابة و الحذفمن برامج خارجة اخرى .

lpSecurityAttributes : مؤشر الى Structure من نوع SECURITY_ATTRIBUTES وهو بارامتر اختيارياذا كان

lpSecurityAttributes يؤشر على nil فان المقبض لن تتم وراثته .

dwCreationDisposition : لتحديد كيفية التعامل مع الملف عند وجوده او عدم وجوده :

1- CREATE_NEW : لإنشاء ملف جديد , اذا كان الملف موجود بالفعل سابقا تفشل الدالة .
2- CREATE_ALWAYS : لإنشاء ملف جديد , اذا كان الملف موجود سابقا يتم مسحه والكتابة فوقه بخصائص الملف الجديد مع اعطاء صفة "أرشيف" للملف الجديد .
3- OPEN_EXISTING : لفتح ملف موجود سابقا , اذا لم يكن الملف موجود تفشل الدالة .
4- OPEN_ALWAYS : لفتح ملف موجود سابقا , اذا لم يكن الملف موجود يتم انشاءه من جديد .
5- OPEN_EXISTING : فتح ملف موجود سابقا إن لم يكن موجود تفشل الدالة .
6- TRUNCATE_EXISTING : لفتح ملف موجود سابقا مع مسح جميع بياناته , لفعل ذلك يجب تحديد العلم GENERIC_WRITE مسبقا , تفشل الدالة اذا لم يكن الملف موجود مسبقا .

dwFlagsAndAttributes : لتحديد صفات الملف :

1- FILE_ATTRIBUTE_ARCHIVE : ملف أرشيف .
2- FILE_ATTRIBUTE_HIDDEN : ملف مخفي .
3- FILE_ATTRIBUTE_NORMAL : ملف عادي .
4- FILE_ATTRIBUTE_READONLY : ملف للقراءة فقط .
5- FILE_ATTRIBUTE_SYSTEM : ملف نظام .
هناك ايضا : FILE_ATTRIBUTE_OFFLINE - FILE_ATTRIBUTE_TEMPORARY - FILE_ATTRIBUTE_COMPRESSED .

hTemplateFile : لتحديد قالب يتم اخذ خصائص الملف منه وهو بارامتر اختياري .

القيمة المرجعة من الدالة : اذا تم استدعاء الدالة بالشكل الصحيح ولم تفشل تكون القيمة المرجعة هي مقبض للملف يمكن استخدامه عن

طريق دوال القراءة والكتابة مثل **ReadFile-WriteFile-CreateFileMapping** .

2- الدالة SetFilePointer :

```
Function SetFilePointer(hFile: THandle; lDistanceToMove: Longint;
  lpDistanceToMoveHigh: Pointer; dwMoveMethod: DWORD): DWORD; stdcall;
```

hFile : مقبض الملف أو المنفذ المفتوح بالدالة **CreateFile** .

lDistanceToMove : عنوان نقل مؤشر القراءة أو الكتابة في الملف .

lpDistanceToMoveHigh : يستخدم كزيادة على البارامتر السابق اذا كان الملف كبير جدا (اكبر من DWORD) وهو بارامتر اختياري

يمكن جعله NIL .

dwMoveMethod : يحدد هذا البرامتر كيفية ازاحة المؤشر ويمكن ان يكون كالتالي :

1- FILE_BEGIN : يتم نقل المؤشر مقدار **lDistanceToMove** من اول الملف = **0** .
2- FILE_CURRENT : يتم نقل المؤشر مقدار **lDistanceToMove** من عنوان المؤشر الحالي = **1** .
3- FILE_END : يتم نقل المؤشر مقدار **lDistanceToMove** من اخر الملف = **2** .

3- الدالة ReadFile/WriteFile:

```
Function ReadFile(hFile: THandle; var Buffer; nNumberOfBytesToRead: DWORD;
  var lpNumberOfBytesRead: DWORD; lpOverlapped: POverlapped): BOOL; stdcall;
```

hFile : مقبض الملف الذي تم فتحه سابقا عن طريق الدالة **CreateFile** , للكتابة يجب ان يكون الملف فتح بالعلم **GENERIC_WRITE** و للقراءة العلم **GENERIC_READ** ويمكن دمج العلمين معا لإتاحة القراءة والكتابة من نفس المقبض .

lpBuffer : عنوان المساحة (بفر) التي سيتم كتابة(قراءة) البيانات منها الى(من) الملف , يجب ان تكون اكبر من او تساوي عدد البايتات المراد كتابتها(قراءتها) .

nNumberOfBytesToWrite-nNumberOfBytesToRead : عدد البايتات المراد كتابتها(قراءتها).

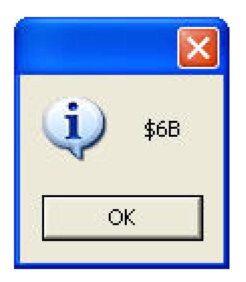
lpNumberOfBytesWritten-lpNumberOfBytesRead : بفر (4 بايت) لإرجاء عدد البايتات التي تم كتابتها(قراءتها) بعد تنفيذ الدالة , يمكنك المقارنة بينها وبين عدد البايتات المراد كتابتها(قراءتها) للتأكد من كتابة (قراءة) كل البفر .

lpOverlapped : مؤشر الى structure من نوع **OVERLAPPED** اذا تم فتح الملف عن طرق العلم **FILE_FLAG_OVERLAPPED** يمكن جعله Nil.



* مثال بسيط قراءة البايت الثالث من ملف نصي :

```
Var CrFileResult:hwnd;
Buf: Byte;
BytesR: Dword;
Begin
CrFileResult := CreateFile('c:\1.txt', GENERIC_READ, FILE_SHARE_READ, nil,
OPEN_EXISTING,
 FILE_ATTRIBUTE_NORMAL, 0);
If CrFileResult <> INVALID_HANDLE_VALUE then
 Begin
 SetFilePointer(CrFileResult, 2, nil, FILE_BEGIN );
 ReadFile(CrFileResult, Buf, 1, BytesR, nil);
 MessageBox(0, PChar('$' + IntToHex(Buf,2)), '', $40 + $2000 + $00);
 CloseHandle(CrFileResult);
 End;
End;
```



و أخيرا في هذا الدرس الجد متواضع أرجوا أن يفيدكم ويسهل عليكم فهم بعض الإشكاليات في ما يخص التعامل مع الملفات بالدلفي فخير الكلام ما قل و دل ... فما هي إلا بعض المعلومات و لك أن تبحث و تزيد عن ذلك و تصحح من أخطائه .. و بالتوفيق للجميع و السلام عليكم و رحمة الله و بركاته .